REMARKS

This is in response to the Office Action of March 4, 2004. Claims 1-21 were rejected. Claims 10, 18, 19, and 20 were amended. Claims 1-21 are pending.

Applicant has made a clarifying amendment to independent claims 10, 18, 19, and 20. This amendment is supported, for example, by original claim 1 and by paragraph [1139], which describes how tracing can be controlled by commands within a program itself through a software settable trace control register.

One aspect of Applicant's claimed invention is that trace controls are included in the compiled program, as described in paragraph [1140] of Applicant's specification. One benefit of Applicant's claimed invention is that the compiled code initiates tracing on its own, without requiring the user's physical presence to initiate tracing using debugger breakpoints or other interactive trace commands, as described in paragraph [1140] of Applicant's specification.

The Examiner rejected claims 1-21 under 35 U.S.C. 102(e) as being anticipated by Torrey et al. (U.S. Pat. No. 6,145,123). Applicant respectfully traverses the rejections.

Torrey utilizes a breakpoint technique in which a programmer sets specific breakpoint addresses in debug registers, as described in column 6, lines 56-65; column 9, lines 60-67; and column 10, lines 29-38. As a program executes, the trace unit checks the address of each instruction for a matching address stored in a debug register, as described in column 10, lines 5-15. The trace unit makes tracing control decisions based upon detecting a match of an address of an instruction with an address stored in a debug register. For example, an "initiate trace address" may be stored in a dedicated "initiate trace breakpoint register" while a "terminate trace address" may be stored in a "terminate trace breakpoint register," as described in column 3, lines 63-67. Tracing is initiated by the trace unit when the program accesses the initiate trace address and tracing is terminated by the trace unit when the program accesses the terminate trace address, as described in column 4, lines 2-5.

Independent claims 1, 6, 10, 18, 19, and 20 include a limitation corresponding to controlling tracing based upon at least one "trace control command" embodied in instructions of a program to be traced. Torrey does not teach or suggest trace control commands that permit a program to initiate tracing from within the program itself. Torrey has no trace control commands embodied within instructions of the program to be traced. Instead, Torrey teaches a breakpoint technique that requires a user to identify and store the addresses at the beginning and the end of each segment of a program to be traced in respective debug registers, as described in column 10, lines 29-45.

Furthermore, in regards to independent claims 10, 18, 19, and 20 Torrey does not have software-settable trace control registers adapted to be set by trace control commands embodied within the program itself. Torrey discloses a system in which a computer programmer may set breakpoint addresses within debug registers of a trace unit. However, the program to be traced does not set these debug registers.

In view of the foregoing amendments and remarks, it is respectfully submitted that the application is now in condition for allowance. The Examiner is invited to contact the undersigned if there are any residual issues that can be resolved through a telephone call.

The Commissioner is hereby authorized to charge any appropriate fees to Deposit Account No. 03-3117.

Dated: May 27, 2004

Respectfully submitted,
COOLEY GODWARD LLP

By: Edw Van

Edward A. Van Gieson
Reg. No. 44,386

Cooley Godward LLP
ATTN: Patent Group
Five Palo Alto Square
3000 El Camino Real
Palo Alto, CA 94306-2155
Tel: (650) 843-5000
Fax: (650) 857-0663